Floyd warshall algorithm

Diagram

Description automatically generated

* 모든 정점에서 정점으로 최단 알고리즘을 구현 하는 것이다

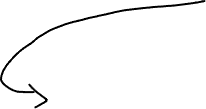
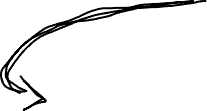
현재까지 계산된 최소 비용

Graphical user interface, application

Description automatically generated

* No immediate connection to vertex 🡪 무한

Calculation 1: 노드 1을 거쳐가는 경우

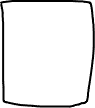


Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated



* 1이 포함되지 않은 경우만 갱신이 가능

Graphical user interface, application, Word

Description automatically generated

Calculation



공식: X 에서 Y로 가는 최소 비용 vs X 에서 노드 1 으로 가는 비용 + 노드 1 에서 Y로 가는 비용

* 2 🡪 3
  + 2 🡪 3 (has cost of 9) (기존에 있던 값)
  + 2 🡪 1 (has cost of 7) + 1 🡪 3 (has cost of 무한)
  + 최소 비용 🡪 9
* 2 🡪 4
  + 2 🡪 4 (has cost of 무한) (기존에 있던 값)
  + 2 🡪 1 (has cost of 7) + 1 🡪 4 (has cost of 8) == 15
  + 최소 비용 🡪 15

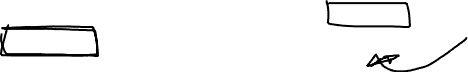
Graphical user interface, application

Description automatically generated



Graphical user interface, application, table, Teams

Description automatically generated



* 3 🡪 2
  + 3 🡪 2 (has cost of 무한) (기존에 있던 값)
  + 3 🡪 1 (has cost of 2) + 1 🡪 2 (has cost of 5)
  + 최소 비용 🡪 7

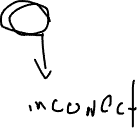
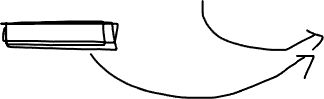
Graphical user interface, application

Description automatically generated



Graphical user interface, application, Teams

Description automatically generated

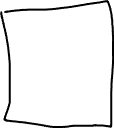
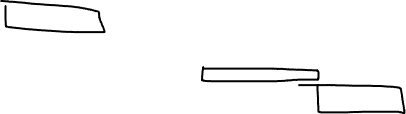


노드 2를 거치는 경우



Table

Description automatically generated



Calculate

* 1 🡪 3
  + 1 🡪 3 (has cost of 무한)
  + 1 🡪 2 (has cost of 5) + 2 🡪 3 (has cost of 9)
    - Has total of 14
  + 최소 비용 🡪 14

Table

Description automatically generated



Table

Description automatically generated



* 1 🡪 4
* 3 🡪 1
* 3 🡪 4
* 4 🡪 1
* 4 🡪 3